

CACHING INFRASTRUCTURE

BACKGROUND

[0001] In a collaborative environment, some computer applications allow multiple clients to simultaneously edit a document. As multiple clients are editing the document, a server may maintain a copy of the document. The server may need to continually update the copy of the document to reflect the current state of the document. This can lead to inefficiencies.

[0002] For example, even with only a single client editing the document, the client's application may need to continuously update the server to allow the transition to multi-party editing when it occurs. This can increase server load, thereby causing a degradation in performance.

SUMMARY

[0003] Example systems and methods described herein relate to a caching infrastructure.

[0004] According to one aspect, an example method for retrieval of presence metadata includes: receiving a request from a first client to edit a document file; sending short-term check out metadata to the first client to begin an editing session; and writing the transition ID to a transition table stored in a cache, wherein the presence of another transition ID in the cache indicates that a document has switched from a single-client mode to a multi-client mode.

[0005] According to another aspect, an example system includes a cache infrastructure for retrieval of presence metadata. The system includes a memory storage unit, and a processing unit coupled to the memory storage unit, wherein the processing unit is operative to receive short-term check out metadata from a first client to begin an editing session, add a transition ID to the short-term check out metadata, write the transition ID to a transition table stored in a cache, and switch from a single-client mode to a multi-client mode. Switching from the single-client mode to the multi-client mode comprises the processing unit being operative to notice when the first client attempts to take another short-term lock on the document and seeing that that a second client has already received the document. The processing unit is operative to determine if the multi-client mode is in progress by checking a database, receive a ping from the first client to determine if the transition ID is in the cache, and when the transition ID is not in the cache, receive pings from the first client at regular intervals, and save the document to the server a plurality of times without incurring any reads/writes to the database, and, when the transition ID is in the cache, receive a ping from the first client to collect a lock table from the database to identify the second client, and receive the second client's lock information by receiving pings that do not carry lock information.

[0006] According to yet another aspect, an example client computer for retrieval of metadata relating to a multi-client editing session includes a memory unit, and a processing unit operative to receive a document from a storage device, the document comprising short-term check out metadata indicating an editing session has begun, ping a cache to determine if another transition ID is stored in the cache, and send a transition ID to a transition table stored in a cache when the another transition ID is not stored in the cache to switch from a single-client mode to a multi-client mode when the another transition ID is stored in the cache.

[0007] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter. Nor is this Summary intended to be used to limit the claimed subject matter's scope.

DESCRIPTION OF THE DRAWINGS

[0008] Non-limiting and non-exhaustive embodiments are described with reference to the following figures, wherein like reference numerals refer to like parts throughout the various views unless otherwise specified.

[0009] FIG. 1 is a schematic block diagram illustrating an example authoring system;

[0010] FIG. 2 is a schematic block diagram illustrating the authoring system of FIG. 1 in which a document stored on first computing device can include content and metadata;

[0011] FIG. 3 is a schematic block diagram of an example lock table;

[0012] FIG. 4 is a schematic block diagram of an example authoring environment including a first computing device on which a master copy of a document to be authored is to be stored;

[0013] FIG. 5 is a schematic block diagram of an example client computing system configured to implement an authoring environment;

[0014] FIG. 6 is a flowchart illustrating an example caching process implemented by an authoring application to recognize a single or multiple clients editing a document; and

[0015] FIG. 7 is a flowchart of an example subroutine used in the method of FIG. 6 for writing a transition ID to a transition table.

DETAILED DESCRIPTION

[0016] In the following detailed description, references are made to the accompanying drawings that form a part hereof, and in which are shown by way of illustrations specific embodiments or examples. While the disclosure will be described in the general context of program modules that execute in conjunction with an application program that runs on an operating system on a computer system, those skilled in the art will recognize that the disclosure also may be implemented in combination with other program modules. The embodiments described herein may be combined and other embodiments may be utilized without departing from the spirit or scope of the present disclosure. The following detailed description is therefore not to be taken in a limiting sense, and the scope of the disclosure is defined by the appended claims and their equivalents.

[0017] Embodiments of the present disclosure provide an environment in which multiple clients can collaboratively author a document while consuming minimal server resources. In example embodiments, when a coauthoring capable application is editing a document, the application adds coauthoring metadata to the document and keeps a server copy of the file updated with the latest metadata. The coauthoring metadata and server copy allows seamlessly transitions from a single client to a multi-client editing state and vice versa. To keep conflicts to a minimum, the coauthoring metadata is uploaded to the server. When a new client opens the document, the client is notified of the areas of the document where other clients are working.